



Interactivité efficace pour les algorithmes évolutionnaires interactifs. Application au débruitage multifractal d'images

Evelyne Lutton, Mario Pilz, Jacques Lévy Véhel

► To cite this version:

Evelyne Lutton, Mario Pilz, Jacques Lévy Véhel. Interactivité efficace pour les algorithmes évolutionnaires interactifs. Application au débruitage multifractal d'images. [Rapport de recherche] RR-5535, INRIA. 2005, pp.20. inria-00070472

HAL Id: inria-00070472

<https://inria.hal.science/inria-00070472>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

***Interactivité efficace pour les algorithmes
évolutionnaires interactifs. Application au
débruitage multifractal d'images.***

Evelyne LUTTON — Mario PILZ — Jacques LEVY VEHEL

N° 5535

Mars 2005

_____ Thème COG _____

A large blue rectangle occupies the lower half of the page. Overlaid on it is a large, light gray stylized 'R' logo. To the right of the 'R', the words 'Rapport de recherche' are written in a white serif font. A horizontal gray brushstroke is positioned below the text.

***Rapport
de recherche***



Interactivité efficace pour les algorithmes évolutionnaires interactifs. Application au débruitage multifractal d'images.

Evelyne LUTTON , Mario PILZ*, Jacques LEVY VEHEL

Thème COG —Systèmes cognitifs
Projet Complex

Rapport de recherche n° 5535 —Mars 2005 — 20 pages

Résumé : Les algorithmes évolutionnaires interactifs pâtissent souvent de ce que l'on peut appeler le "goulot d'étranglement de l'utilisateur", autrement dit une perte d'efficacité de l'algorithme due à la fatigue de l'utilisateur (qui n'apprécie évidemment pas des interactions répétitives et/ou inintéressantes). Nous proposons et analysons dans ce papier une méthode de gestion de l'interaction-utilisateur permettant à la fois de glaner les informations utiles au bon déroulement de l'algorithme d'optimisation évolutionnaire, tout en limitant le nombre d'interactions. Cette méthode a été développée dans le cadre d'une application en débruitage multifractal d'images : le débruitage multifractal est adapté aux images complexes, mais dépend d'un jeu de paramètres délicats à ajuster. Une approche évolutionnaire interactive simple a été élaborée dans des travaux précédents pour l'ajustement de ces paramètres. Nous étudions ici l'emploi d'un module d'approximation du jugement de l'utilisateur, par le biais d'une "carte de fitness," afin de réduire le nombre d'interactions avec l'utilisateur. La méthode peut être étendue facilement à d'autres applications évolutionnaires, interactives ou calculatoirement coûteuses.

Mots-clés : Algorithmes évolutionnaires, évolution interactive, interactivité, analyse multifractale d'images, exposant de Hölder, ondelettes, spectres multifractals.

* and Dresden University of Technology, Germany

Efficient interactivity for interactive evolutionary algorithms. Application to multifractal image denoising.

Abstract: Interactive evolutionary algorithms (IEA) often suffer from what is called the “user bottleneck.” In this paper, we propose and analyse a method to limit the user interactions, while still providing sufficient informations for the EA to converge. The method has been currently developed on a multifractal image denoising application: a multifractal denoising method is adapted to complex images, but depends on a set of parameters that are quite difficult to tune by hand. A simple IEA has been developed for this purpose in a previous work. We now experiment an approximation of the user judgment, via a “fitness map”, that helps to reduce the number of user-interactions. The method is easily extensible to other interactive, or computationally expensive, evolutionary schemes.

Key-words: Evolutionary algorithms, interactive evolution, user-interaction, multifractal image analysis, Hölder exponents, wavelets, multifractal spectrum.

1 Introduction

The founding works in interactive evolution [3, 2, 4, 1], oriented towards artistic applications, have now been extended to many other applications where quantities to be optimised are related to subjective rating (visual or auditive interpretation). Characteristic attempts are for example [5] for Hearing Aids fitting, [7] for smooth, human-like, control rules design for a robot arm, or [8] for the design of HTML style sheets. An overview of this vast topic can be found in [9].

Interaction with humans raises several problems, mainly linked to the “user bottleneck” [11], i.e. the human fatigue. Several solutions have been considered [11, 9, 10] :

- reduce the size of the population and the number of generations,
- choose specific models to constrain the research in a priori “interesting” areas of the search space,
- perform an automatic learning (based on a limited number of characteristic quantities) in order to assist the user and only present to him the most interesting individuals of the population, with respect to previous votes of the user.

In the application of interest in this work, a first experiment has been done according to the first of the previous items (evolving a small population). We now experiment an approach related to the third item, i.e. we try to extend the fitness rating to individuals of a larger population via the analysis of the user judgment on a small sample of individuals.

The paper is organised as follows : In section 2 the basics of multifractal image analysis and denoising are recalled. In section 3 the first interactive method is presented, then the proposed extension are described in section 4. Experiments are described and commented in section 5, showing the efficiency of the proposed method. Conclusions, possible extensions, and future work are detailed in section 6.

2 Multifractal bayesian denoising

2.1 Multifractal image analysis

Multifractal analysis relies on the hypothesis that important information of the signal are embedded in its irregularity. This approach is adapted to complex and/or noisy signals.

In image analysis, it consists in measuring the regularity at each sample point, in grouping the points having the same irregularity, and then in estimating the Hausdorff dimension (i.e. the “fractal dimension”) of each iso-regularity set.

Irregularity is measured via the local Hölder exponent [12] defined for a continuous (non-differentiable) function f at x_0 as the largest real α such that :

$$\exists C, \rho_0 > 0 : \forall \rho < \rho_0 \quad \sup_{x, y \in B(x_0, \rho)} \frac{|f(x) - f(y)|}{|x - y|^\alpha} \leq C$$

The function f_H that gives for an α the Hausdorff dimension of its iso- α set, $f_H(\alpha)$, is called a multifractal spectrum. It is thus a representation of the irregularity of the image (a sort of measurement of the geometrical distribution of the iso- α sets). As an example, a $f_H(\alpha) \simeq 1$ corresponds

to a linear and smooth structure, while $f_H(\alpha) \simeq 0$ is a set of scattered points (singular points), or $f_H(\alpha) \simeq 2$ is a uniformly textured area.

The multifractal spectrum provides at the same time a local (α) and a global ($f_H(\alpha)$) view-point on data. It has been exploited with success in many applications where irregularity bears some important information (image segmentation [13], signal and image denoising [12], etc ...)

Wavelet transforms are convenient tools for the estimation of the Hölder exponents. The method we present here is based on discrete wavelet transforms, and has been compared to other denoising techniques based on wavelets (soft thresholding), known as very efficient in many cases, see [14].

2.2 Regularity based bayesian denoising

The aim is to find a denoised image I_2 for a noisy image I_1 , under the constraint that I_2 has a given multifractal spectrum. Furthermore, we assume that the noise is white and gaussian. Its variance is denoted σ .

Following the Bayesian MAP approach from [15] we denote by y a wavelet coefficient of the noisy image at scale j . The corresponding wavelet coefficient \hat{x} of the denoised image at the same scale j can be calculated by solving the following equation :

$$\hat{x} = \arg \max_{x > 0} \left(j \cdot g \left(\frac{\log_2 (\hat{K} \cdot x)}{-j} \right) - \frac{(|y| - x)^2}{2\sigma^2} \right) \text{sgn}(y)$$

where

- \hat{K} is a constant for which $\hat{K} \cdot |y| < 1$ holds and may be set independently for each scale. In what follows, \hat{K} has been taken as the inverse of the maximal coefficient in each scale j .
- g is a function which defines the multifractal spectrum of the denoised image. It is determined by the 5 values $\alpha_{\min}, \alpha_{\text{nod}}, \alpha_{\max}, g(\alpha_{\min})$ and $g(\alpha_{\max})$. The spectrum has been chosen to fulfill the following constraints :
 - g is defined on the interval $[\alpha_{\min}, \alpha_{\max}]$,
 - $g(x) \in [0, 1]$,
 - $\alpha_{\text{nod}} \in [\alpha_{\min}, \alpha_{\max}]$ and $g(\alpha_{\text{nod}}) = 1$,
 - g is affine on $[\alpha_{\min}; \alpha_{\text{nod}}]$ and on $[\alpha_{\text{nod}}; \alpha_{\max}]$.

In most cases, but not necessarily, the multifractal spectrum calculated from the denoised coefficients \hat{x} should show a slight spectral shift to the right. This shift is a sign of an overall increase of regularity.

2.3 Free parameters

Consequently, the extended multifractal denoising algorithm presented in this work depends on 7 parameters :

- the 5 values defining the a priori spectrum g ,

- the variance σ of the noise,
- the wavelet used for the discrete (inverse) wavelet transformation.

The choice of the wavelet is less critical than the choice of the other 6 parameters. Usually Daubechies 6 to 12 offer equivalent denoising results in terms of visual reception whereas Daubechies wavelets with smaller supports yield unsatisfactory results in some cases.

Specially in cases where we want to treat very noisy image and subsequently have to set the parameters σ and α_{nod} to relatively high values, the denoising algorithm leads to artefacts in the denoised image when using wavelets with a small support, see figure 1. The regularity of those wavelets is low. They are therefore not able to model very irregular parts of an image.

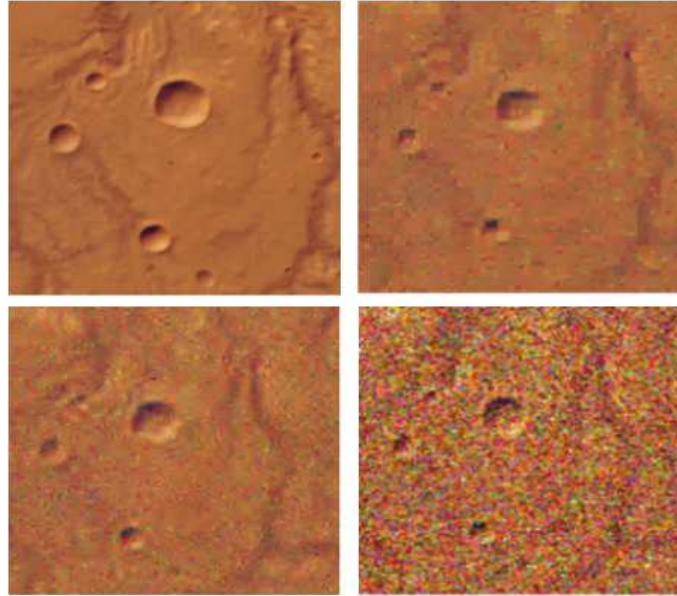


FIG. 1 – **Original Image without noise** (u.l.), Multifractal denoising using wavelet **Daubechies 2** (u.r.), Multifractal denoising using wavelet **Daubechies 18** (l.l.), **Noisy Image** (l.r.), all parameters except wavelets are constant.

It should be mentioned that the number of calculated wavelet scales is fixed to a value obtained from the image dimensions $[N \times M]$:

$$scales = \lfloor \log_2(\max(N, M)) \rfloor$$

The setup of the 7 resulting free parameters is nontrivial in the sense that they are strongly dependent on the amount of noise in the noisy image and the subjective opinion of the human observer about which result reflects best the desired denoised image.

The idea was therefore to build an user guided evolutionary algorithm to interactively find suitable settings of the free parameters.

3 A First Interactive Approach

An Interactive Evolutionary Algorithm for multifractal denoising has already been developed in the INRIA - Complex Team. The results of this work have been presented in [14].

3.1 Genome

The first denoising IEA does not include the choice of the wavelet basis as a free parameter but considers a shift to the a priori spectrum g for diagonal wavelet coefficients. It has been noticed that the diagonal wavelet coefficients are more sensitive towards additive noise and therefore may deserve a different spectrum g .

The genomes that are evolved by this IEA are made of 7 real genes :

- 5 values to define the g function for the horizontal and vertical wavelet coefficients : $\alpha_{min} \in [0, 0.5]$, $g(\alpha_{min}) \in [0, 1]$, $\alpha_{nod} > \alpha_{min}$, $\alpha_{nod} \in [0, 2]$, $\alpha_{max} > \alpha_{nod}$, $\alpha_{max} \in [0.01, 20]$, $g(\alpha_{max}) \in [0.2, 1]$
- the shift of the g function for the diagonal coefficients (range $[0, 0.5]$),
- the variance of gaussian noise, σ (range $[0, 100.0]$).

3.2 Fitness and user interaction

The fitness function is given by the user with aid of sliders attached to each denoised image. Possible evaluations are in the range $[-10, +10]$, where -10 is the worst and +10 is best possible notation to an image. The default value of “0” corresponds to a denoised image which seems neither to be better nor worse than the original noisy image.

Beside controlling the evolutionary cycle by giving evaluations to images, the user may also directly edit the genotypes of these images, see figure 2. This means that the user can behave as an additional genetic operator.

3.3 Genetic Engine

The population has a fixed size of 6 individuals. Each individual carries a set of 7 parameters and therefore represents one solution for the afore mentioned combinatorial problem. All individuals are presented as an image, the result of our denoising algorithm. The basic evolutionary cycle employed in this program is illustrated in figure 3.

The evolutionary operators are defined in the following way :

- **Parent Selection** is performed by deterministic selection of the 3 best individuals in the population.

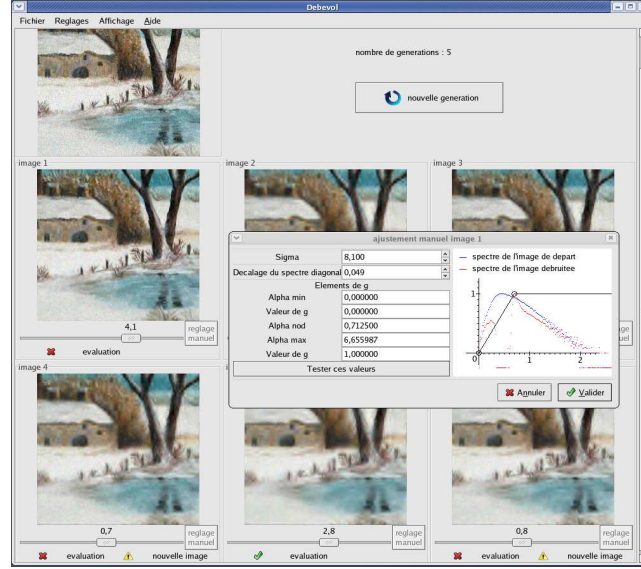


FIG. 2 – The interface of the previous IEA, written in C++.

– **Genetic Operators :**

- **Barycentric Crossover**, where a new individual is a weighted combination of his parents with a randomly chosen weight in $[0, 1]$.
- **Mutation** as an independent perturbation of each gene value by adding a gaussian noise with a given variance.
- **Survivor Selection** replaces the 3 worst individuals in the parent generation by the offspring individuals.

Before parent selection, the user given fitness values are weighted with a sharing value to maintain a minimum of diversity inside the small population. This sharing value is calculated from mean genotype distances. The parent selection then chooses the 3 individuals with the best weighted fitness and is therefore fully deterministic. Crossover and mutation operators then produce an offspring of size 3. The survivor selection puts the offspring in the place of the parent individuals and thereby closes the evolutionary cycle.

4 Extended Interactive Approach

The main issue of the present work is to build an interactive evolutionary algorithm that takes more benefit from the user evaluations (increases the reactivity) while being able to handle popula-

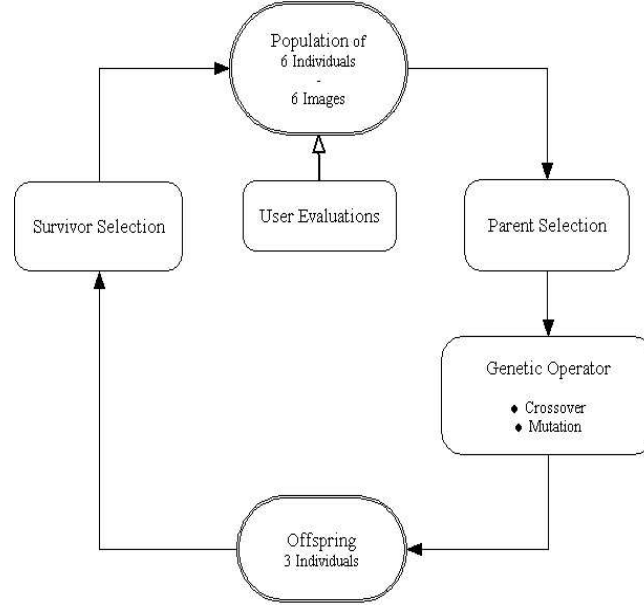


FIG. 3 – The first denoising IEA genetic engine.

tions of any size (increases the search capabilities). The first denoising IEA is based on a population which contains only 6 individuals. Furthermore, it only considers the current user evaluations for the calculation of the next generation. In this basic scheme, the user has access to 6 individuals (or images) per generation : the IEA is driven by a fitness sample – or let us say *a fitness map* – made of only 6 points.

A dynamic approximation of the interactive fitness is a delicate task, and necessitates a rather large sample. We propose a method based on the use of past user notations, collected in a set, the *fitness map*. The fitness of new individuals produced by the genetic engine can be preliminary estimated from the fitness map by smooth interpolation (flat or polynomial, see in section 4.3). This preliminary fitness estimation can serve as a preselection tool in order to show to the user only the 6 best individuals of a larger current population.

The use of larger population sizes offer some major advantages, where an obviously easier maintenance of diversity, a more extensive exploration of the given search space and a possible speedup of convergence are the most significant.

4.1 Genome

The genomes that are evolved by this extended IEA are made of 7 genes :

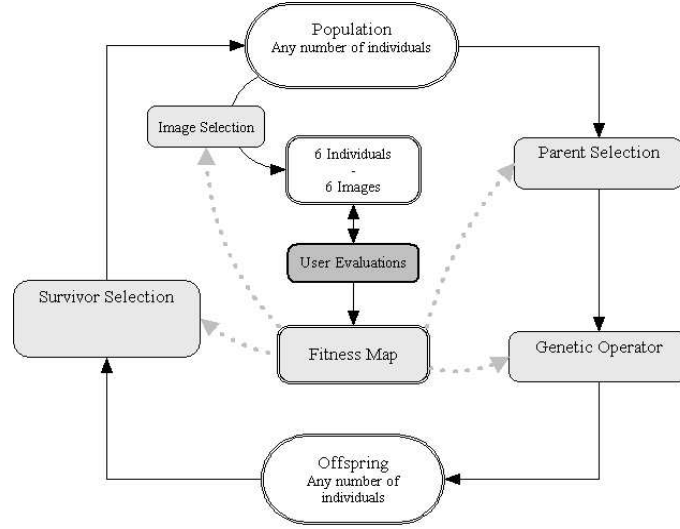


FIG. 4 – **The extended genetic engine.** It supports a fitness map (illustrated in this figure) and complex user interaction.

- 5 values to define the g function for the horizontal and vertical wavelet coefficients : $\alpha_{min} \in [0, 0.5]$, $g(\alpha_{min}) \in [0, 1]$, $\alpha_{nod} > \alpha_{min}$, $\alpha_{nod} \in [0, 2]$, $\alpha_{max} > \alpha_{nod}$, $\alpha_{max} \in [0.01, 20]$, $g(\alpha_{max}) \in [0.2, 1]$
- the wavelet used for the discrete wavelet transformation (Daubechies 2 to 20),
- the variance of gaussian noise, $\sigma \in [0, 100]$.

4.2 Fitness and user interaction

User evaluations to the images are given in almost the same way as presented in 3.2 : The user gives his notations with help of slides attached to the images within the range $[-6(\text{verybad}), \dots, 0(\text{neutral}), \dots, +6(\text{verygood})]$.

The genetic engine of the extended IEA is highly customisable by setting parameters. In contrast to the behavior of the first IEA, it is possible that all 6 images in the user interface are changed from a generation to the next. But it might obviously be frustrating for the user to loose images that he maybe considered as interesting. Therefore, the possibility to mark images as “Super Individuals” has been introduced. Super individuals remain continuously visible as an image in the user interface and also remain constant as individual in the population. The user may toggle this state at any time by pressing a button, see figure 5.

To increase the variety of user interactivity, two new dialogs have been created :

- a dialog to view and manipulate the individuals in the population (figure 6),

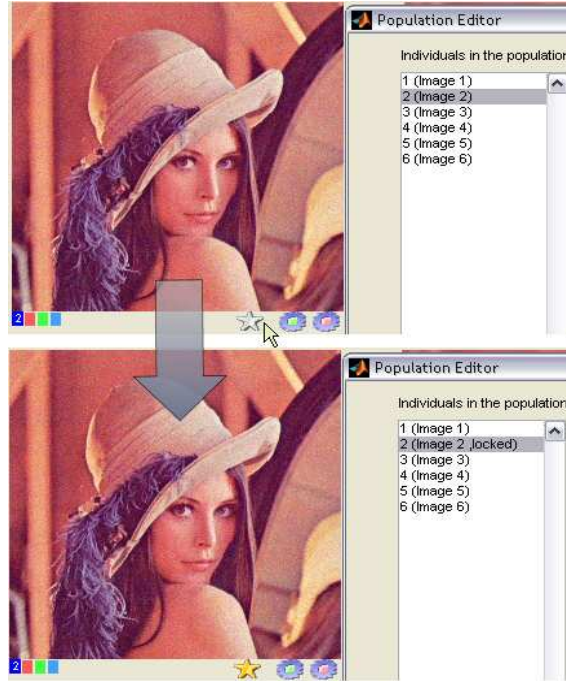


FIG. 5 – Clicking the star button toggles an individual as “Super Individual”. From thereon it is treated as constant in the population.

- a dialog to view and manipulate the samples in the fitness map (figure 7).

These 2 dialogs both provide plots of the gene values of the individuals in the population, respectively fitness map, along with their corresponding fitness values. By toggling checkboxes, additional curves, such as an interpolation of fitness and sharing values, are shown.

An other tool of user interactivity was created by introducing “user ranges” (figure 8). These ranges are soft-thresholds that constrain the search space of genes and can be set independently for each of the 7 genes. The effects of these ranges on the genes of individuals are described in 4.3.

The new IEA also includes a history function which allows to recall the state of the population at any time.

4.3 Genetic Engine

The introduction of a varying population size and a fitness map required some major changes to the genetic engine of the first IEA. The extended IEA therefore differs from the first IEA in many points. Selection methods now strongly depend on the fitness map. Crossover and mutation operators may also depend on the fitness map. The genetic cycle was extended by an “image selection”. This

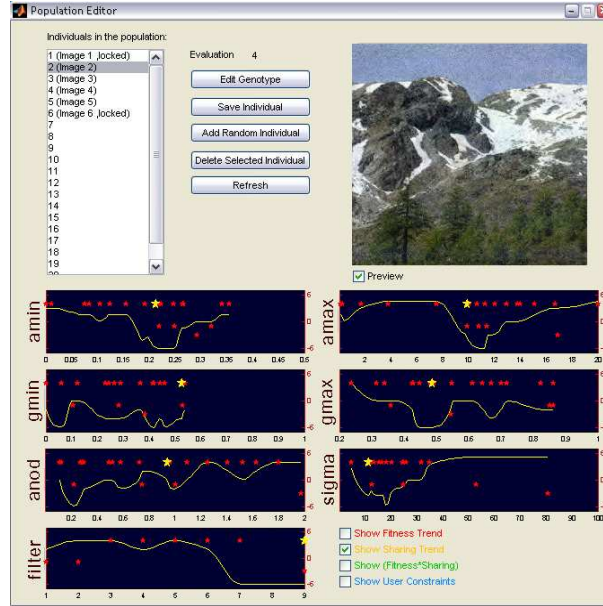


FIG. 6 – **The population editor.** Individuals may be added to the population, existing individuals may be deleted and their genotype can be manipulated. The gene values are plotted on 7 curves as red stars. The plotted curves are interpolations of the fitness (or shared fitness) samples values.

method selects the 6 individuals that are shown to the user as images. The extended evolutionary cycle employed in this program is illustrated in figure 4.

The fitness map is a matrix of size $[8 \times N]$. N is the number of samples that are saved in the fitness map. These samples are vectors of size $[8 \times 1]$, made of a genotype and its corresponding fitness value. The fitness map is used to interpolate between the available samples in order to predict the fitness values of unknown genotypes. Two interpolation methods have been implemented :

- **“nearest”** : The fitness value of the nearest sample in the fitness map is returned as the fitness value of the unknown sample.
- **“interpolation”** : Interpolating polynomials of order 8 are calculated for each gene using the samples of the fitness map (red stars interpolated by red lines in figure 9). The approximated fitness value for an unknown sample (see green markers in figure 9) is the mean value of the 7 polynomials for the genes values of the unknown sample.

Various selection algorithms have been implemented. These selection operators can be deployed by the parent-, offspring- and image selection. The respective selection operator that is actually used in a certain stage of the genetic cycle is set offline with help of a configuration file.

The available selection methods are the following :

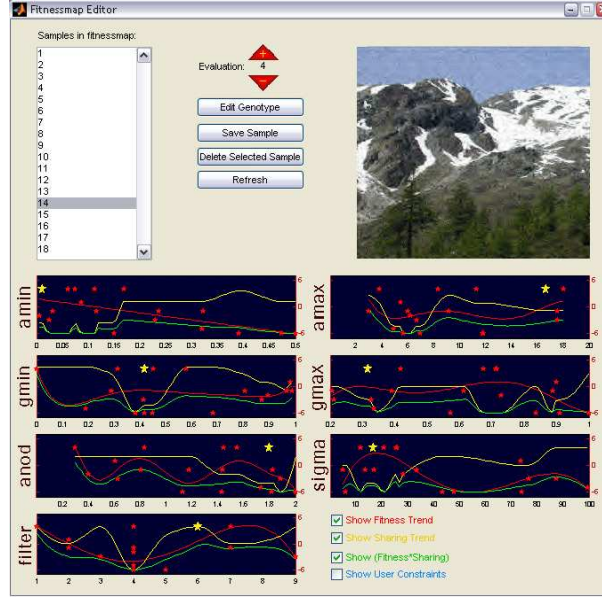


FIG. 7 – The fitness map editor. Samples may be deleted and their fitness can be reevaluated.

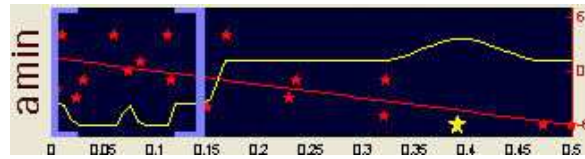


FIG. 8 – Plot of sample fitness for the values of α_{min} . An interpolation of the fitness values is plotted as a red curve. The sharing estimation is plotted as a yellow curve. Setting an **user preferred range** for individual genes is done by drag and drop of the blue brackets.

- “**fit test**” : The Individual with the best fitness value is selected.
- “**cycle**” : n individuals are selected by cycling through a number m of the fittest individuals. This method can be used to generate an offspring from a small number of parent individuals (as in the first IEA).
- “**roulette**” : Randomized variant of fitness-proportionate selection.
- “**rank**” : Randomized variant of rank-proportionate selection. The selection probability for an individual is $pressure^{-rank}$, where “pressure” adjusts the strenght of selection and “rank” is the position of the individual inside the population (sorted by decreasing fitness values).

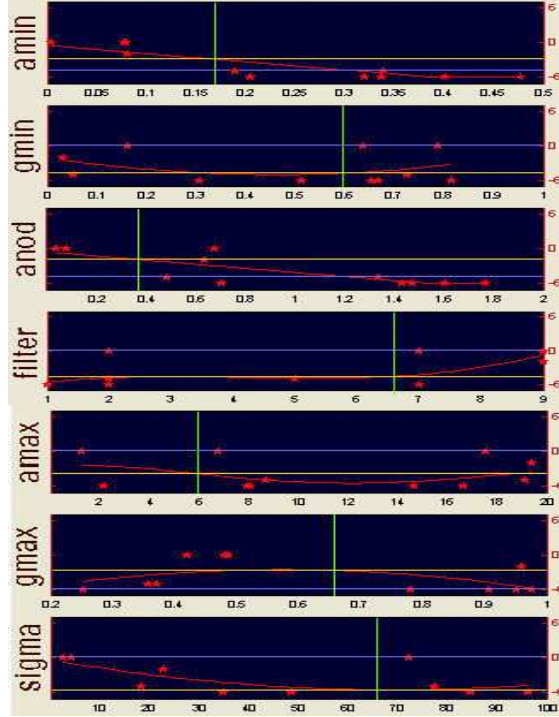


FIG. 9 – The 2 fitness estimation methods illustrated for a sample genotype (green) : **nearest method** (blue), **interpolation method** (yellow). The y axes represent the fitness values, while the x axes represent the gene values

A sharing algorithm has been implemented. Equally to the sharing algorithm of the first denoising IEA, fitness values are weighted with a sharing factor that is calculated from mean genotype distances inside the population. Genotypes with a high mean distance to the other genotypes in the population consequently have a bigger gain of fitness. The pressure of this sharing method can be set in a configuration file, independently for each selection method. Distinct selection of individuals is also implemented and configurable.

Different versions of the genetic operators (crossover and mutation) have been implemented :

- **crossover** : **“random”** : New individuals are a weighted combination of their parents. The weights are randomly chosen in $[0, 1]$. **“swap”** : Special case of random crossover. Parent genes are randomly swapped to generate children genotype. **“factory”** : This methods builds new genotypes out of the best genes from two parent individuals. The necessary fitness for individual genes is taken from the earlier described interpolating polynomials.

- **mutation** : “**random**” : Gaussian perturbation of each gene with a given σ . “**preferred area**” : Gaussian perturbation of each gene towards its user range (4.2). There is no effect on a gene when it is already located inside the user area.

5 Experiments

Quantitative evaluations are rather difficult to perform on interactive evolutionary algorithms. To be able to evaluate the efficiency of the fitness map scheme or, to some extent, compare the first IEA with the extended IEA, experiments were made in a non-interactive way.

The two algorithms were run on several noisy images, for which the initial “non-noisy” images were available, and for various parameter settings.

5.1 The non-interactive software

For these tests, the software was slightly modified (see modified user interface in figure 10). The user evaluations were replaced by automatic evaluations. A user fitness is therefore imitated by the calculation of a phenodistance between the noisy images and their corresponding original images. The two presented versions of the IEA had to run 30 generations on every noisy image and for every parameter setting. In each generation the minimum phenodistance was collected in order to produce a convergence curve. This was repeated for at least 30 times. Afterwards a mean curve of convergence was calculated. The 2 presented versions of the IEA have been compared on the basis of these average curves.

Unfortunately, neither the calculation of an overall pixel color difference nor the calculation of a mean pixel color difference between 2 images can properly modelize the human reception of relevant differences between 2 images. However, we have chosen to use a mean difference to calculate the phenodistance between 2 images.

5.2 Parameters

As these experiments consume a remarkable amount of calculation time, we decided to concentrate on a varying population size for the extended IEA.

The most important parameters used for the tests with the new IEA are :

- Population size : 16, 32, 64 and 128 Individuals.
- Parent selection : Rank selection (as presented in 4.3).
- Offspring size : 90% of parent generation.
- Image selection : Fittest selection.
- Fitness map interpolation : nearest.
- Use of “Super Individual” : in each generation the image with the lowest phenodistance to the original image is set as super individual.

On the basis of the previous parameters, one generation is equivalent to 5 user interactions.

The parameters for the “basic” IEA remained unchanged :

- Population size : 6 Individuals

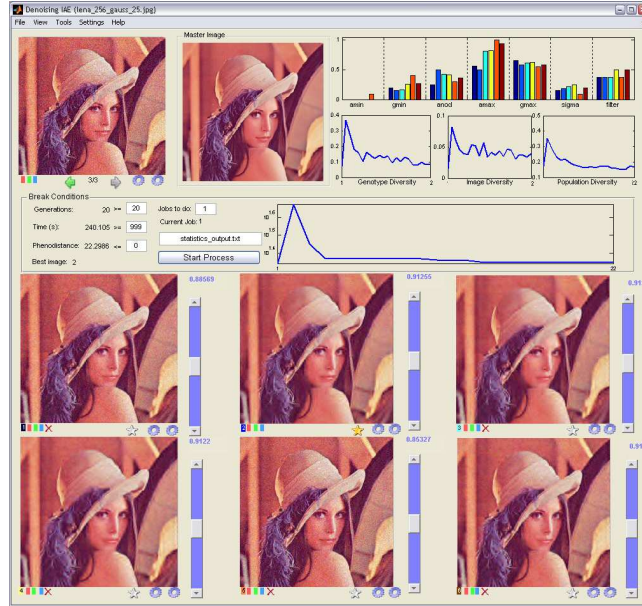


FIG. 10 – The modified interface allows batch jobs.

- Parent selection : Fittest 3,
- Offspring size : 3 Individuals,
- Image selection : not necessary.

On the basis of the previous parameters one generation is equivalent to 3 user interactions.

5.3 Results

To ensure a fair comparison between the two algorithms, the average curves of convergence are plotted with respect to the number of user interactions (i.e. user evaluations) instead of the generations number.

Figures 11, 12 and 13 show a clear improvement of the minimisation behaviour for the fitness map scheme, the larger the population, the more efficient.

The loss of precision of the fitness calculation based on the fitness map, which is a very rough approximation of the user – or phenotypic (for the automated version) – fitness, is compensated by the exploration capabilities of a larger population.

This improved exploration capability has also been noticed in a qualitative manner by users on the original interactive IEA.

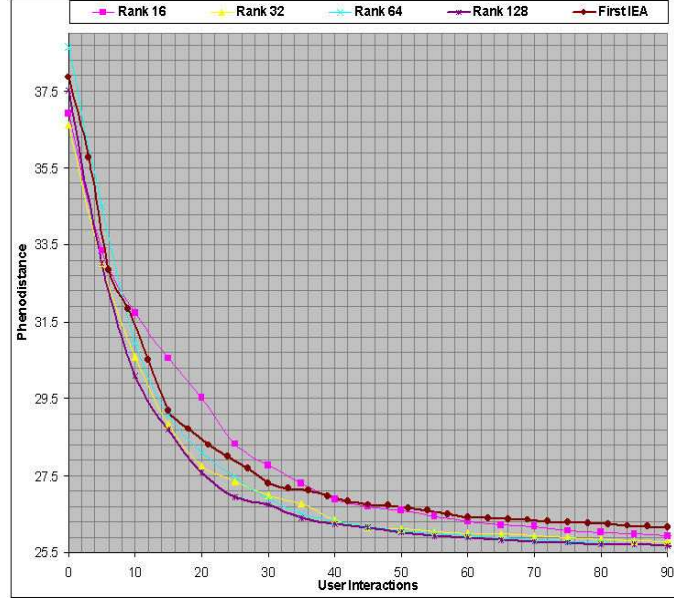


FIG. 11 – Comparison of mean convergence for different population sizes. Original Image : Sommet 256. Noisy Image : Sommet 256 with Gauss $\sigma = 20$.

6 Conclusion

The fitness map scheme has been proved to be efficient on an interactive multifractal image denoising application. The manipulation of a much larger population in conjunction with the use of rough approximations of the user fitness provides a solution to the “user bottleneck” problem. The fitness map scheme can be easily generalised to other application, including non-interactive ones where exact fitness calculations are computationally expensive.

Further work will include extensions of this scheme to other interactive evolutionaru image analysis and signal processing tasks. We also intend to consider the fitness map on difficult non-interactive inverse problems such as the inverse problem for IFS, related to fractal compression techniques.

The presented IEA was originally developed as standalone program. Fully retaining its standalone capabilities, the software has been extended to integrate with the Fraclab Toolbox, see figure 14. It is now easily possible to exchange images between the IEA and the Fraclab Toolbox. The Fraclab Toolbox, including the presented new IEA, is available as download at <http://fractales.inria.fr>.

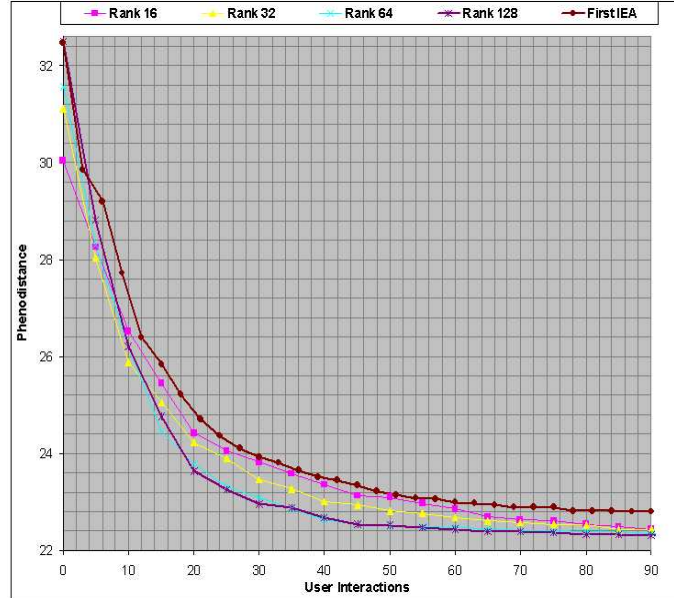


FIG. 12 – Comparison of mean convergence for different population sizes. Original Image : Lena 256. Noisy Image : Lena 256 with Gauss $\sigma = 25$.

Acknowledgment

The authors are grateful to Pierre Grenier, Yann Landrin-Scheitzer and Pierrick Legrand for their technical support, and help to integrate the software into Fraclab.

Références

- [1] P. J. Angeline, “Evolving Fractal Movies”, *Genetic Programming 1996 : Proceedings of the First Annual Conference*, John R. Koza and David E. Goldberg and David B. Fogel and Rick L. Riolo (Eds), pp 503–511, 1996.
- [2] K. Sims, “Interactive evolution of dynamical systems,” in *First European Conference on Artificial Life*, pages 171–178, 1991. Paris, December.
- [3] K. Sims, “Artificial Evolution for Computer Graphics,” *Computer Graphics*, 25(4) :319–328, July 1991.
- [4] S.J.P. Todd and W. Latham, “Evolutionary Art and Computers”, Academic Press, 1992.
- [5] H. Takagi, M. Ohsaki, “IEC-based Hearing Aids Fitting”, *IEEE Int. Conf. on System, Man and Cybernetics (SMC’99)*, Tokyo, Japan, vol 3, pp 657–662, Oct. 12–15, 1999.

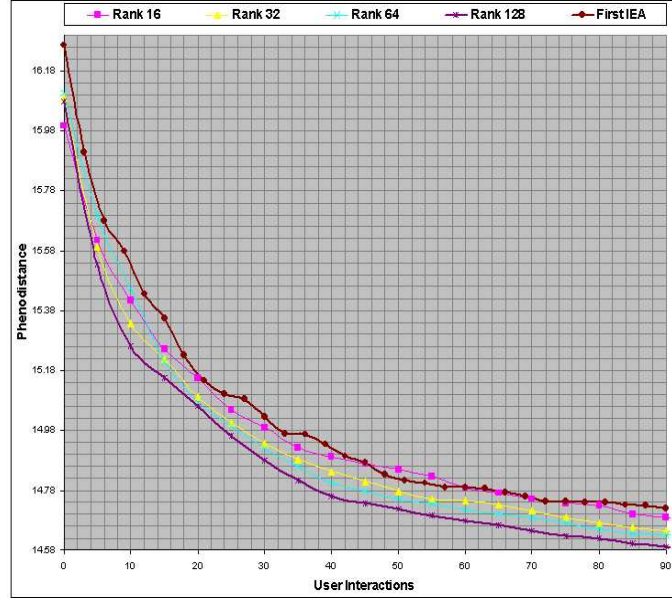


FIG. 13 – Comparison of mean convergence for different population sizes. Original Image : Mars 256. Noisy Image : Mars 256 with Gauss $\sigma = 30$.

- [6] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” in *Soft Computing*, vol 9, pp 3-12, 2005.
- [7] S. Kamohara, H. Takagi and T. Takeda, “Control Rule Acquisition for an Arm Wrestling Robot,” in *IEEE Int. Conf. on System, Man and Cybernetics (SMC’97)*, vol 5, Orlando, FL, USA, pp 4227-4231, 1997.
- [8] N. Monmarche, G. Nocent, G. Venturini, and P. Santini. “On Generating HTML Style Sheets with an Interactive Genetic Algorithm Based on Gene Frequencies”, *Artificial Evolution, European Conference, AE 99*, Dunkerque, France, November 1999, Selected papers, Springer Verlag, LNCS 1829, 1999, C. Fonlupt and J. K. Hao and E. Lutton and E. Ronald and M. Schoenauer (Eds).
- [9] H. Takagi, “Interactive Evolutionary Computation : System Optimisation Based on Human Subjective Evaluation”, *IEEE Int. Conf. on Intelligent Engineering Systems (INES’98)*, Vienna, Austria, pp 1-6, Sept 17-19, 1998.
- [10] W. Banzhaf, “Interactive Evolution”, in *Handbook of Evolutionary Computation*, 1997, Oxford University Press.
- [11] R. Poli and S. Cagnoni, “Genetic Programming with User-Driven Selection : Experiments on the Evolution of Algorithms for Image Enhancement”, in *2nd Annual Conf. on Genetic Pro-*

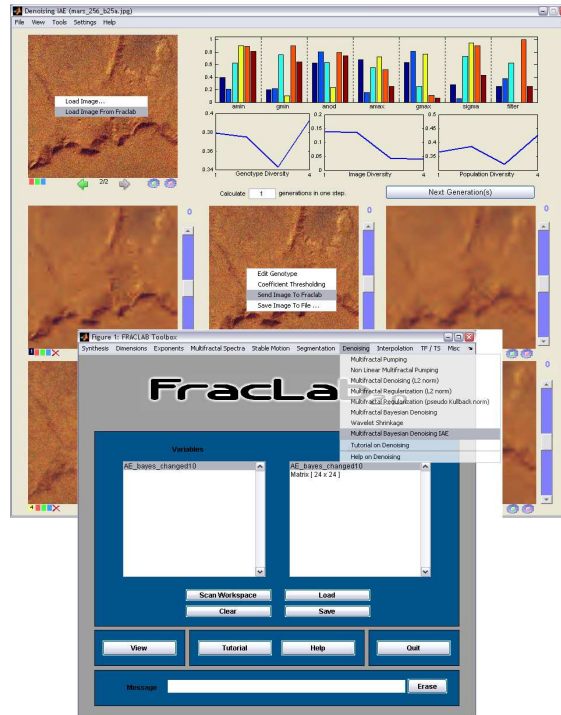


FIG. 14 – The presented IEA integrated with Fractalab.

gramming, pp 269-277, 1997.

- [12] Lévy Véhel, Jacques and Lutton, Evelyne “Evolutionary signal enhancement based on Hölder regularity analysis”, EVOIASP2001 Workshop, Como Lake, Italy, Springer Verlag, LNCS 2038, 2001.
- [13] Lévy Véhel, J. “Fractal Image Encoding and Analysis”, Springer Verlag, 1997
- [14] Evelyne Lutton, Pierre Grenier and Jacques Levy Vehel “An Interactive EA for Multifractal Bayesian Denoising”, 2005.
- [15] Jacques Levy Vehel and Pierrick Legrand “Bayesian Multifractal Signal Denoising”, 2003,
- [16] J. Chapuis and E. Lutton, “ArtiE-Fract : Interactive Evolution of Fractals”, 4th International Conference on Generative Art, Milano, Italy, December 12-14, 2001.
- [17] Y. Landrin-Schweitzer, P. Collet and E. Lutton, “Interactive GP for Data Retrieval in Medical Databases,” in EUROGP’03, 2003.
- [18] E. Lutton, P. Collet, Y. Landrin-Schweitzer and T. Prost, “Introducing Lateral Thinking in Search Engines with InteractiveEvolutionary Algorithms,” in Annual ACM Symposium on

Applied Computing (SAC 2003), Special Track on "Computer Applications in Health Care" (COMPAHEC 2003), March 9 to 12, Melbourne, Florida, U.S.A., 2003.

- [19] S. Rooke, "The Evolutionary Art of Steven Rooke" <http://www.azstarnet.com/~srooke/>
- [20] Y. Semet, Y. Jamont, R. Biojout, E. Lutton and P. Collet, "Artificial Ant Colonies and E-Learning : An Optimisation of Pedagogical Paths," in HCII'03 - Human Computer Interaction International, 2003.



Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399